(12) UK Patent Application (19) GB (11) 2 201 854 (13) A

(43) Application published 7 Sep 1988

(21) Application No 8803143

(22) Date of filing 11 Feb 1988

(30) Priority data
(31) 8703136 (32) 11 Feb 1987 (33) GB

(71) Applicant
University College Cardiff Consultants Limited

(Incorporated in United Kingdom)

P O Box 78, Cardiff, CF1 1XL

(72) Inventors
David Roger Bull
David Heaton Horrocks

(74) Agent and/or Address for Service
Withers & Rogers
4 Dyer's Buildings, Holborn, London, EC1N 2JT

(51) INT CL⁴
H03H 17/00

(52) Domestic classification (Edition J):
H3U 50C1D F

(56) Documents cited
GB 1476603 GB 1409681

(58) Field of search
H3U
Selected US specifications from IPC sub-class
H03H

(54) Digital filter architecture

(57) In a method of digitally filtering an electrical signal or signals, an output signal or signals representing the combination of the products of samples of the input signal or signals and a plurality of multiplying coefficients are generated solely by elementary addition, subtraction and/or binary shift operations. The sequence and interrelationship of the operations is such that a plurality of intermediate or partial results are obtained some of which are each used in the generation of signals representing more than one of the products whereby fewer operations are required than would be necessary if each product was formed separately. Minimisation of the number of operations is achieved by heuristic techniques.

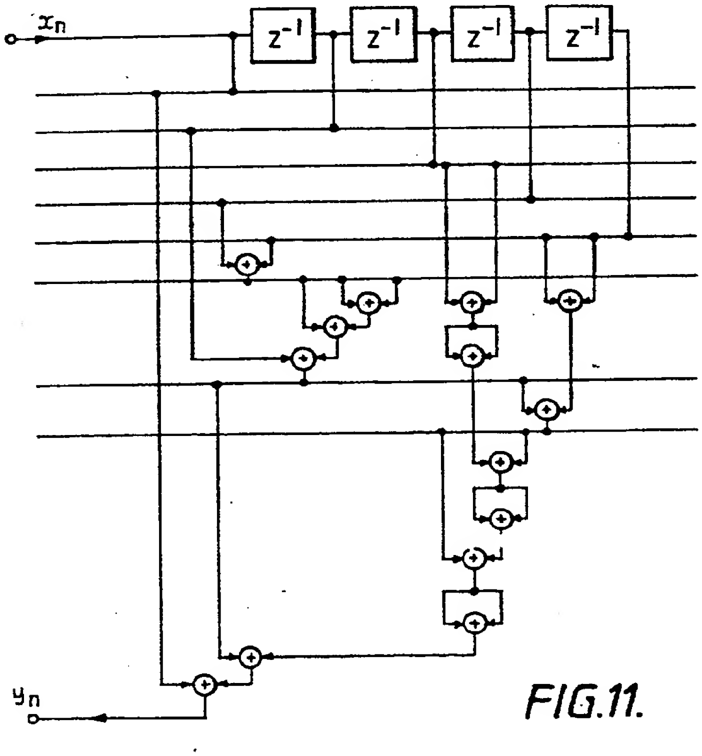


FIG.11.

The drawing(s) originally filed was (were) informal and the print here reproduced is taken from a later filed formal copy.

GB 2 201 854 A

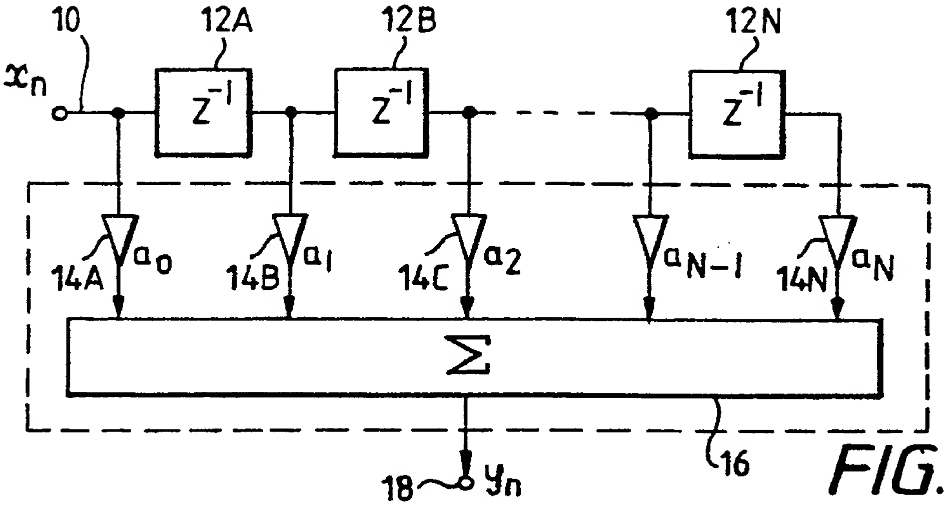


FIG. 1.

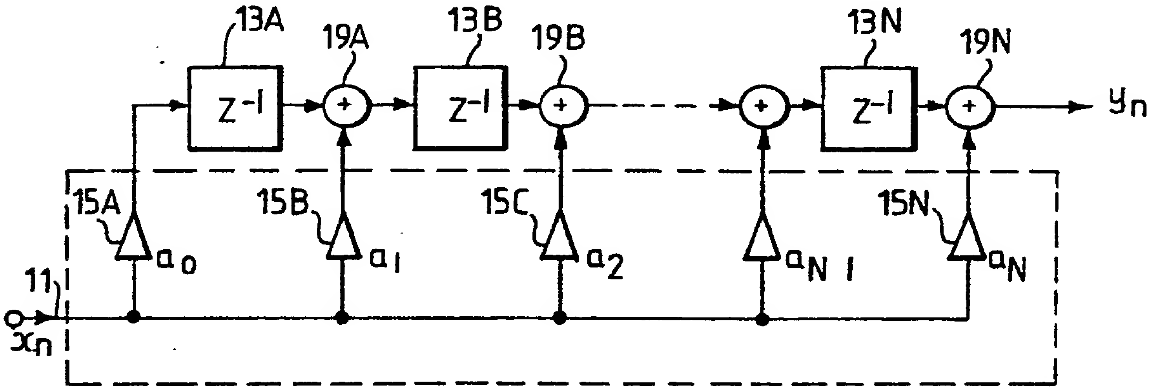


FIG. 2.

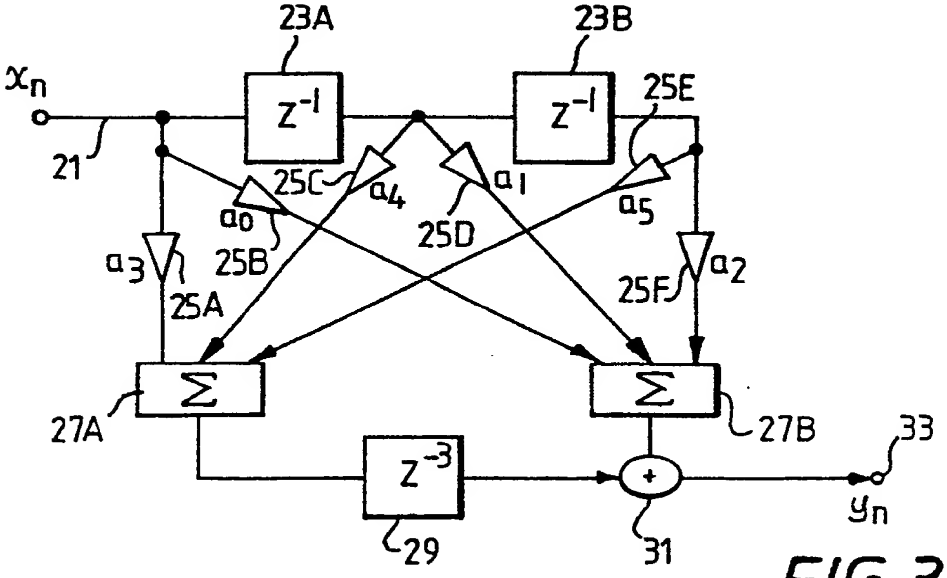


FIG. 3.

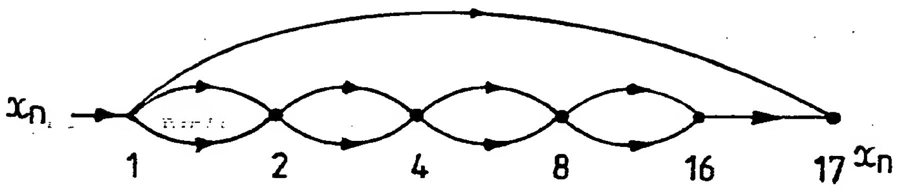


FIG.4.

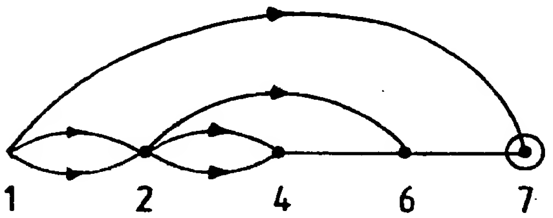


FIG.5A.

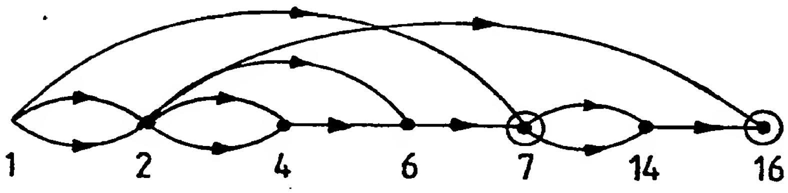


FIG.5B.

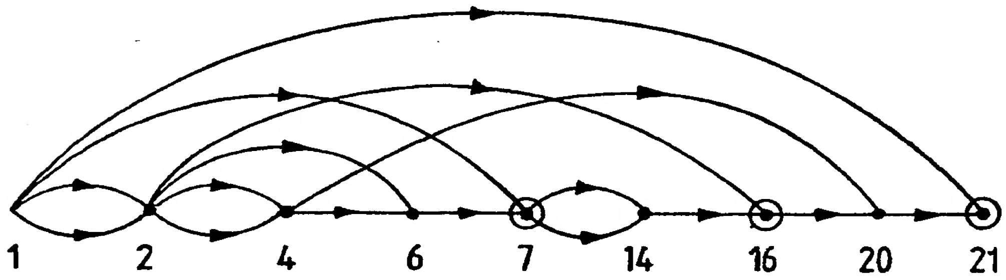


FIG.5C.

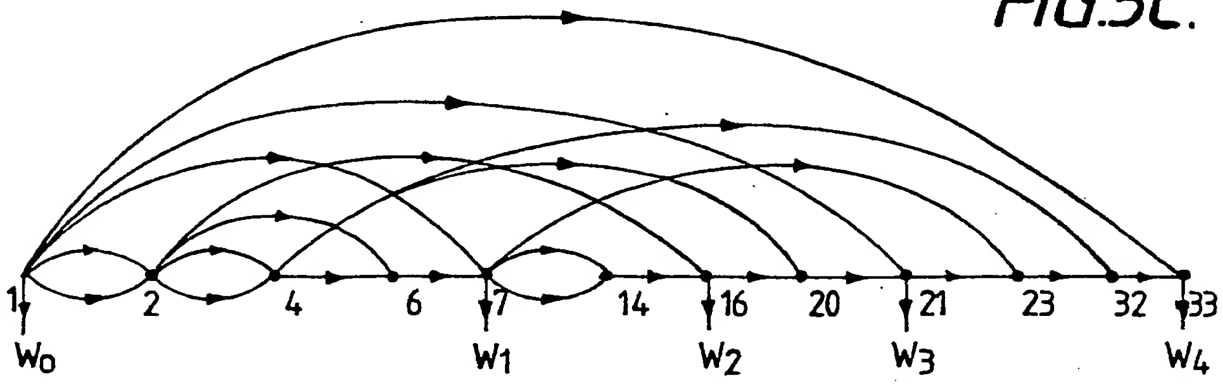


FIG.5D.

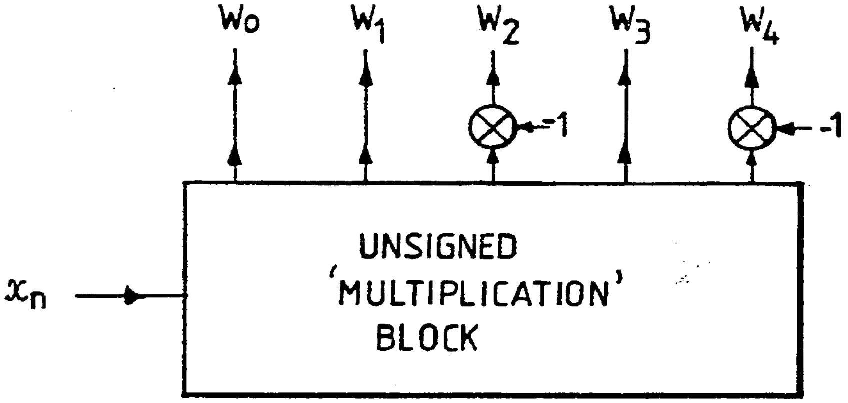


FIG.6.

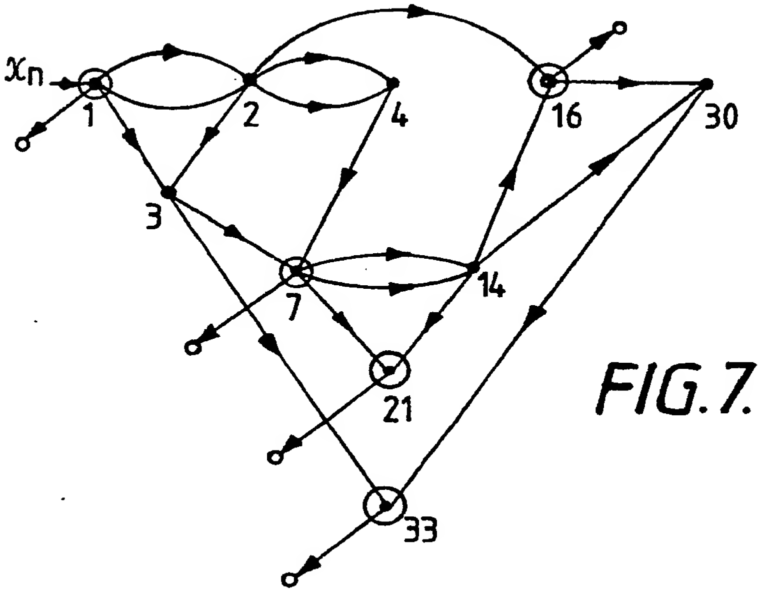


FIG.7.

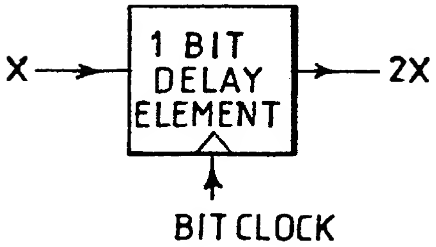


FIG.8A.

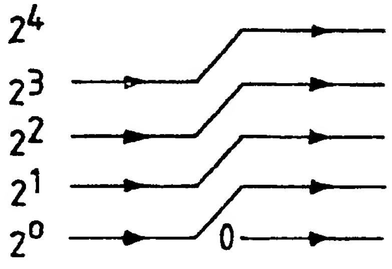


FIG.8B.

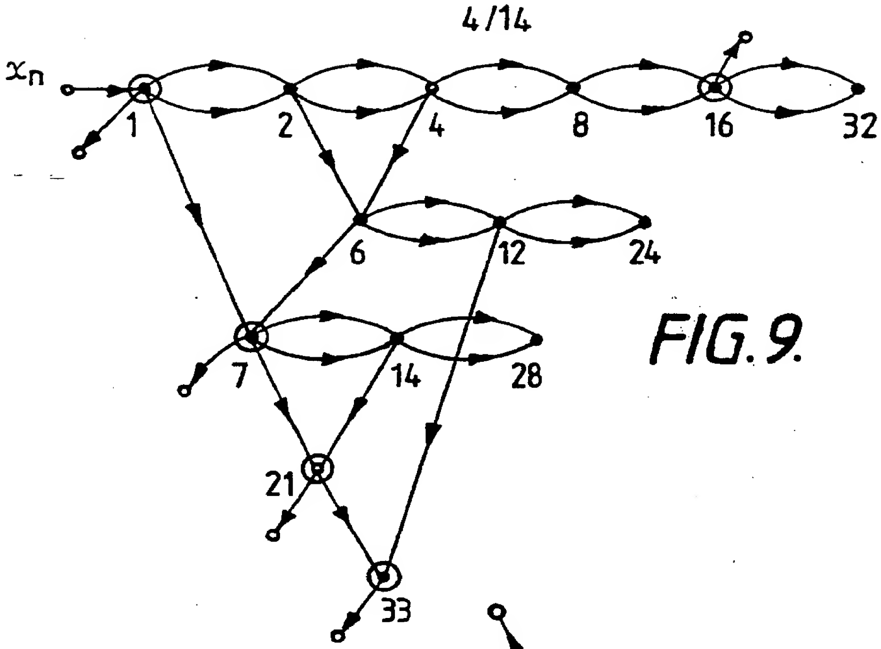


FIG. 9.

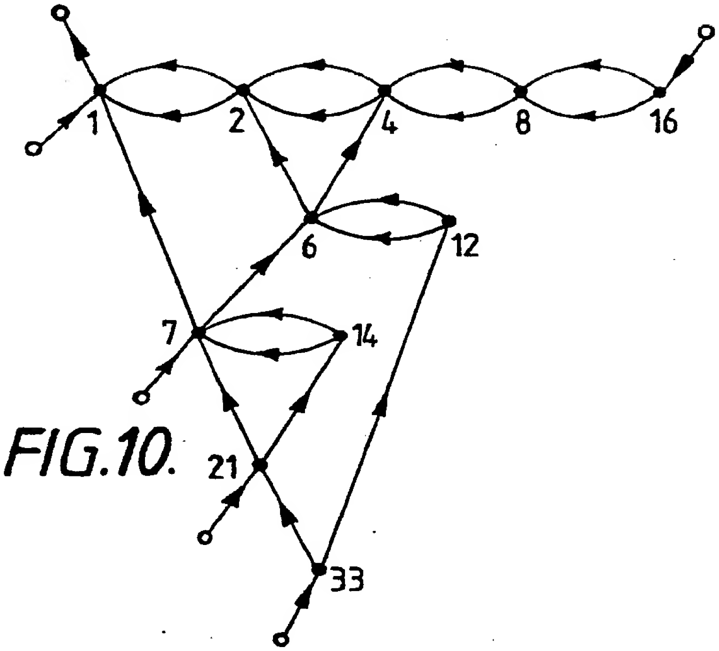


FIG. 10.

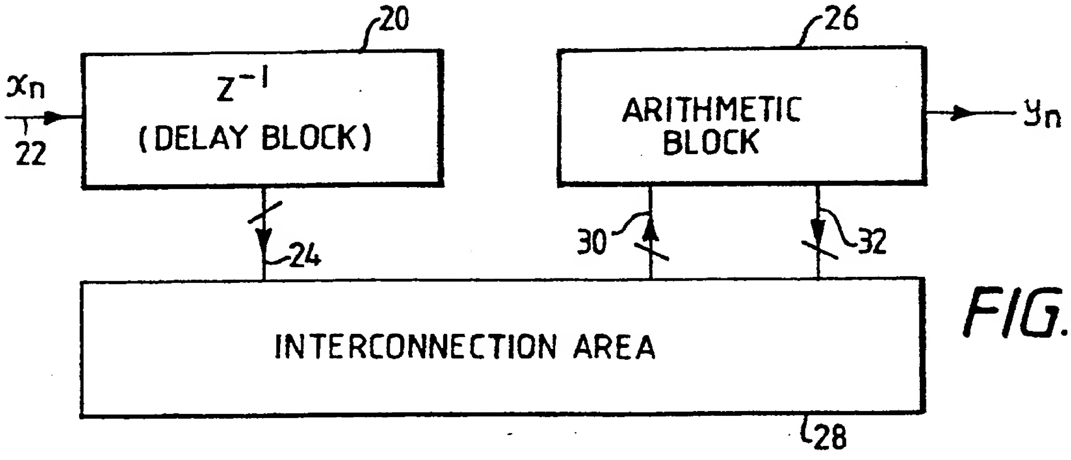


FIG. 13.

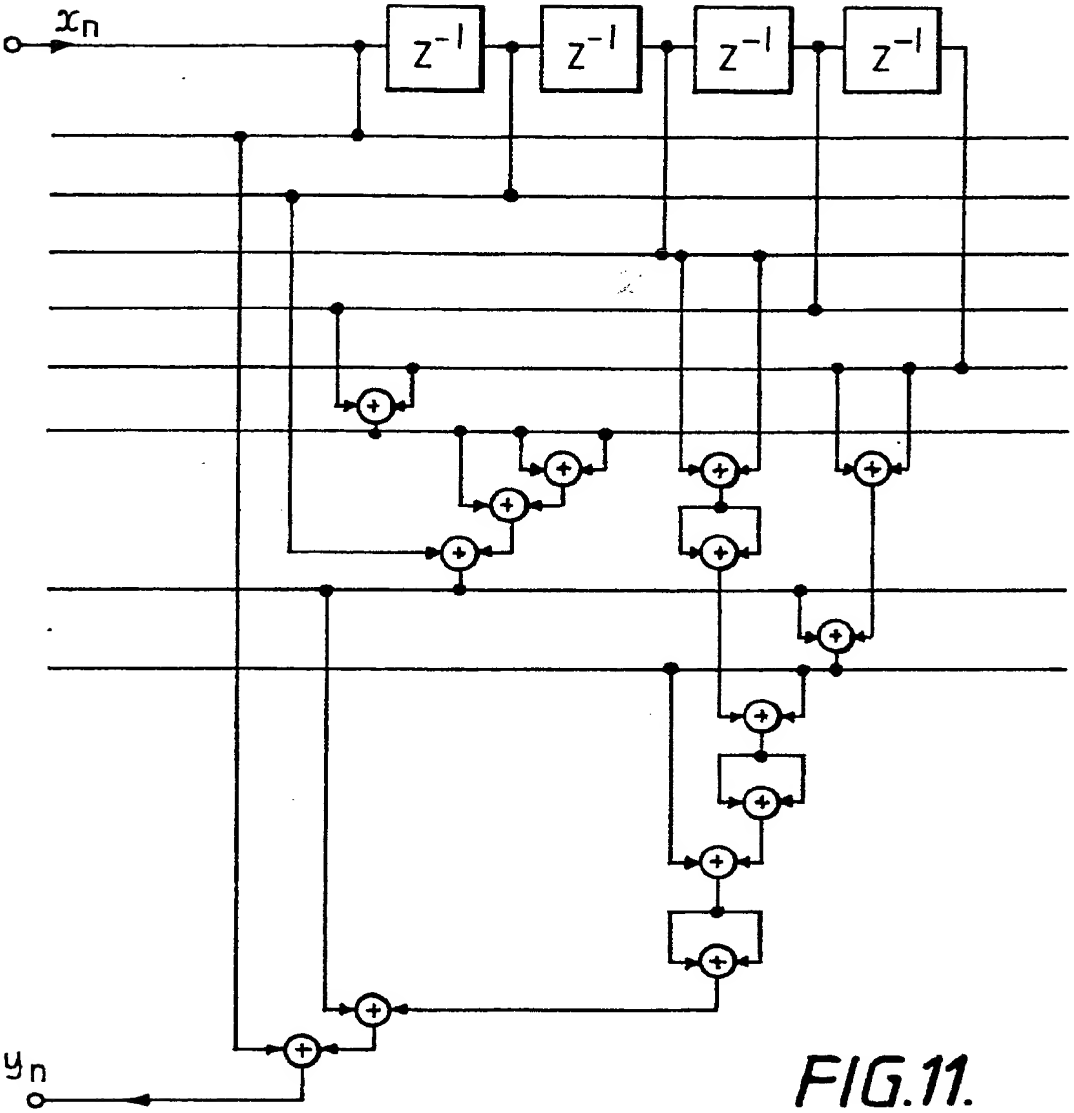
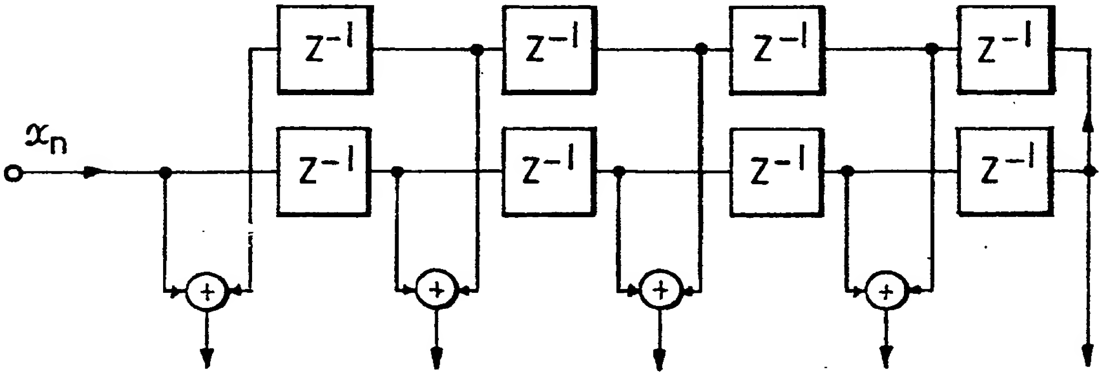


FIG.11.

FIG.12.



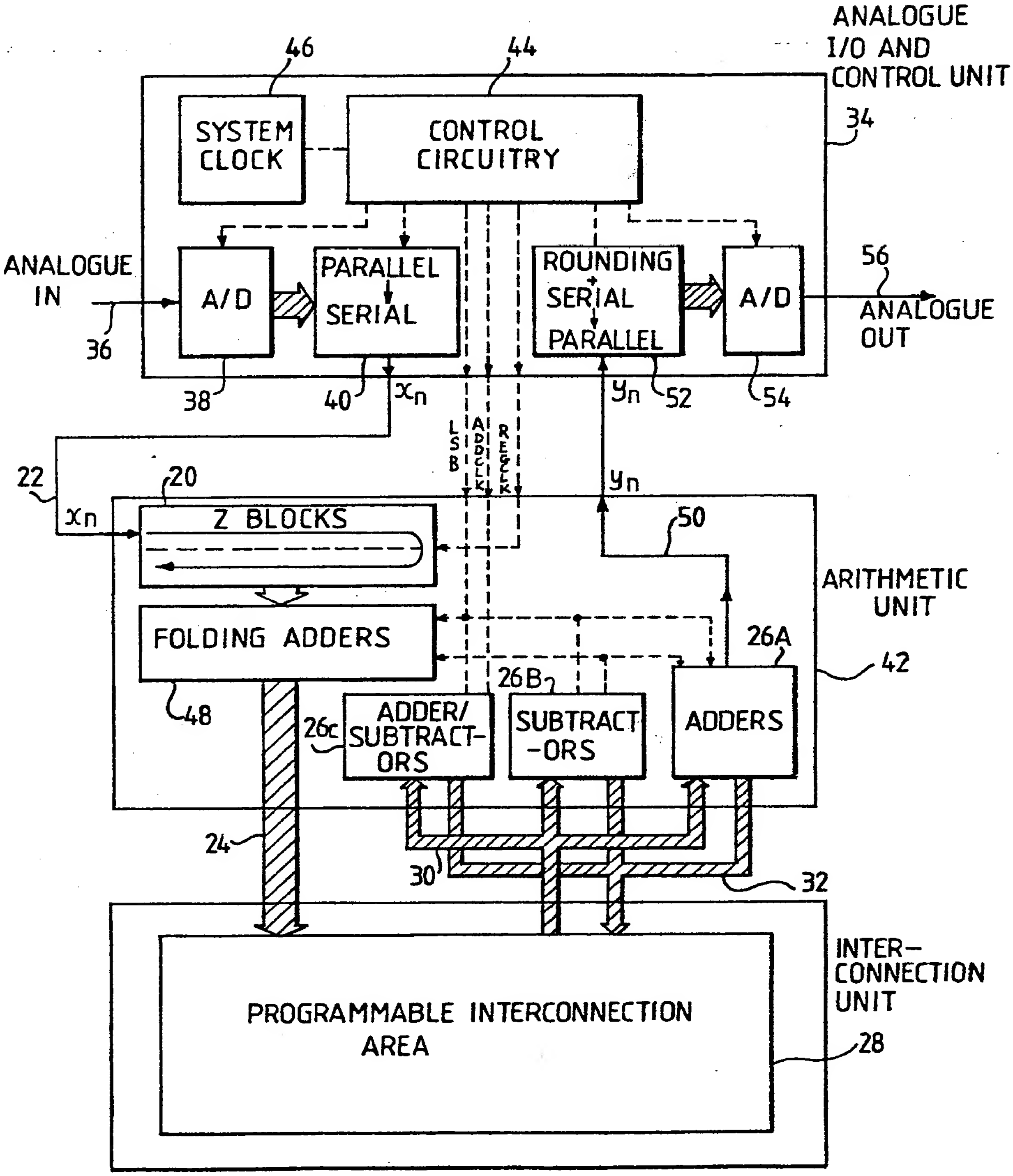
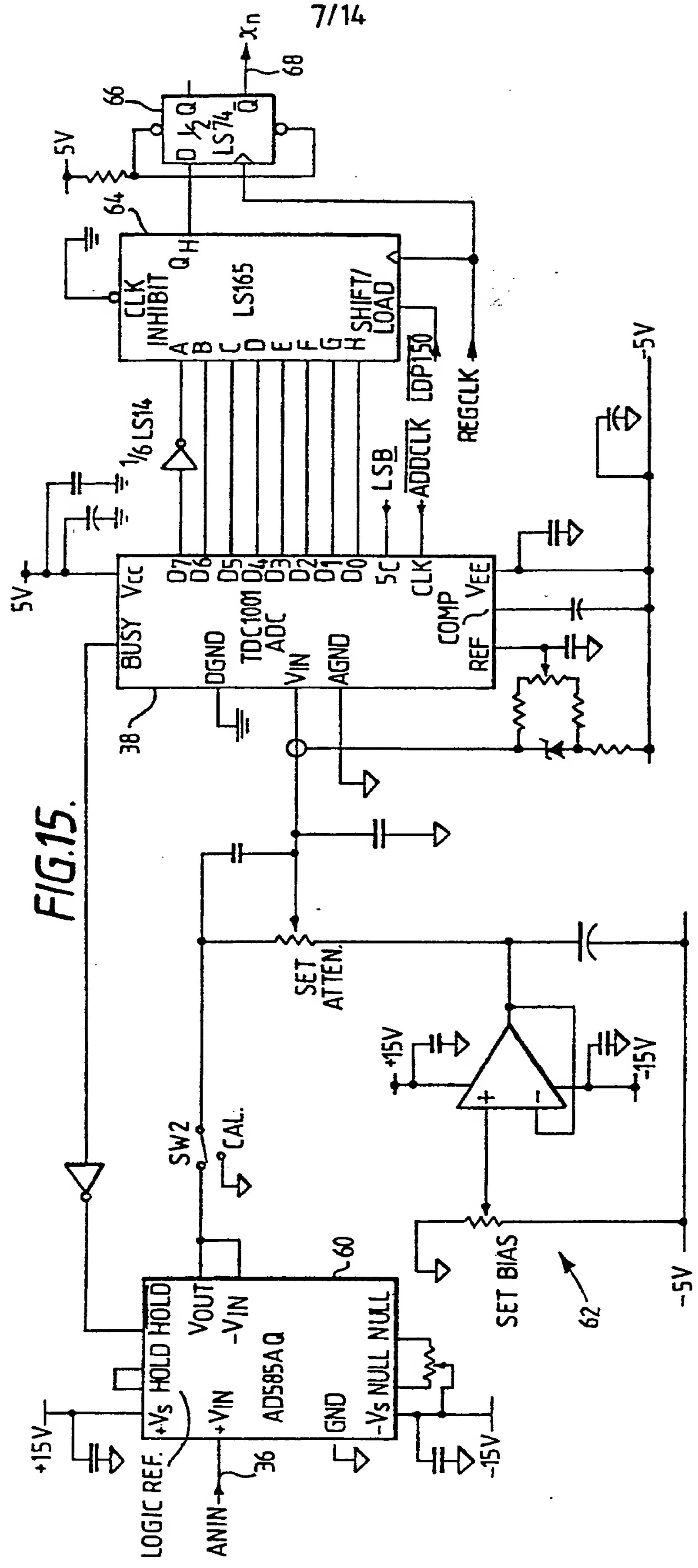

ANALOGUE
I/O AND
CONTROL UNIT
34
46
44
SYSTEM
CLOCK
CONTROL
CIRCUITRY
ANALOGUE
IN
36
A/D
PARALLEL
SERIAL
ROUNDING
SERIAL
PARALLEL
A/D
56
ANALOGUE
OUT
38
40
x_n
y_n
52
54
LSB
ADCLK
REGCLK
22
20
x_n
Z BLOCKS
y_n
50
ARITHMETIC
UNIT
FOLDING ADDERS
26A
42
48
26B
26c
ADDER/
SUBTRACT-
ORS
SUBTRACT
-ORS
ADDERS
24
30
32
INTER-
CONNECTION
UNIT
PROGRAMMABLE INTERCONNECTION
AREA
28

FIG.14.



xn
68
66
D ½ Q
LS74
Q̄
5V
64
CLK INHIBIT
QH
LS165
SHIFT/LOAD
A
B
C
D
E
F
G
H
LDP150
REGCLK
1/6 LS14
LSB
ADDCLK
-5V
5V
Vcc
D7
D6
D5
D4
D3
D2
D1
D0
5C
CLK
VEE
BUSY
DGND
TDC1001 ADC
VIN
AGND
COMP
REF
38
SET ATTEN.
+15V
-15V
SW2
CAL.
60
SET BIAS
62
-5V
HOLD HOLD
VOUT
-VIN
AD585AQ
+Vs
+VIN
GND
-Vs NULL NULL
+15V
LOGIC REF.
36
ANIN
-15V

FIG.15.

FIG.16.

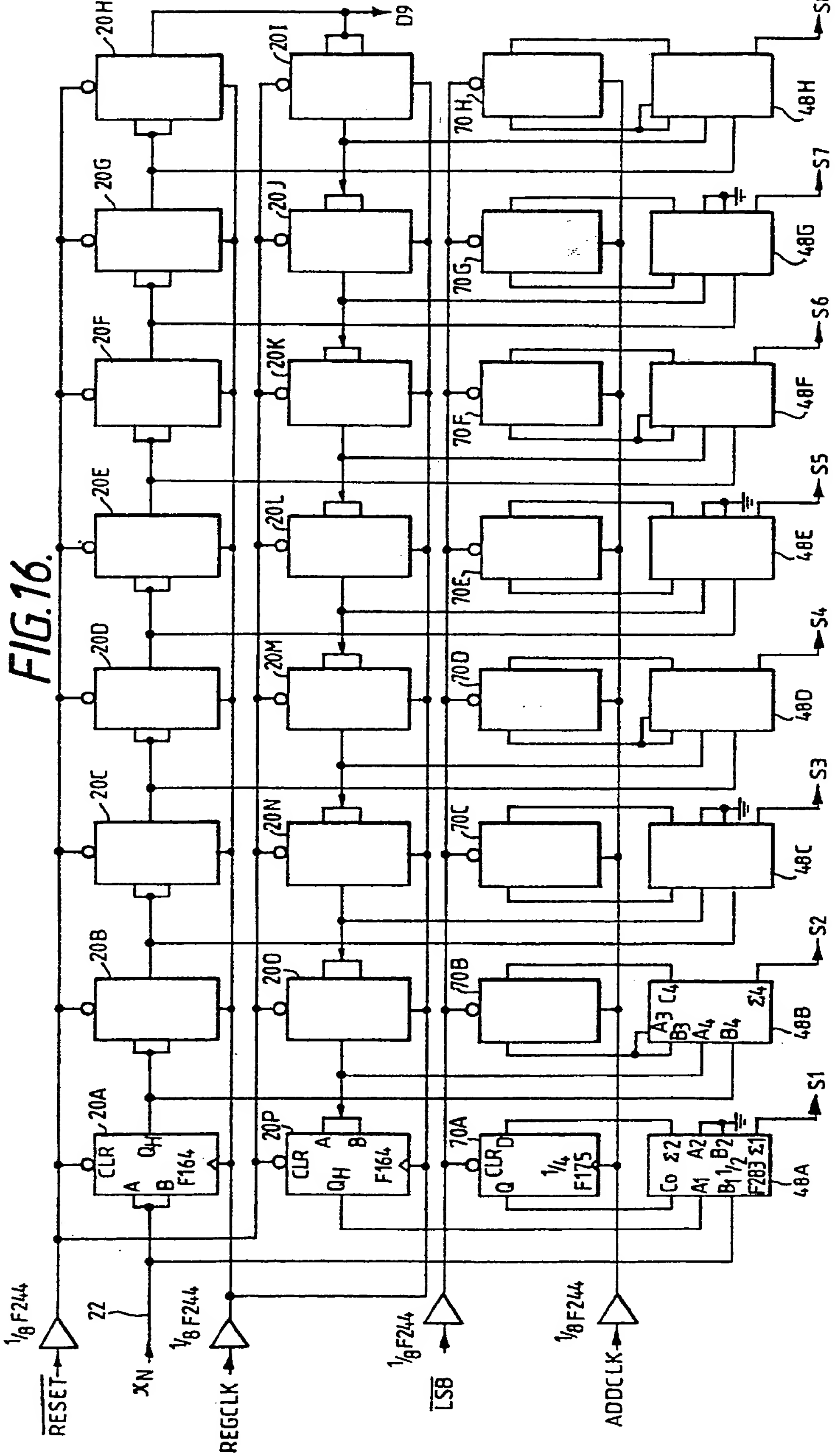
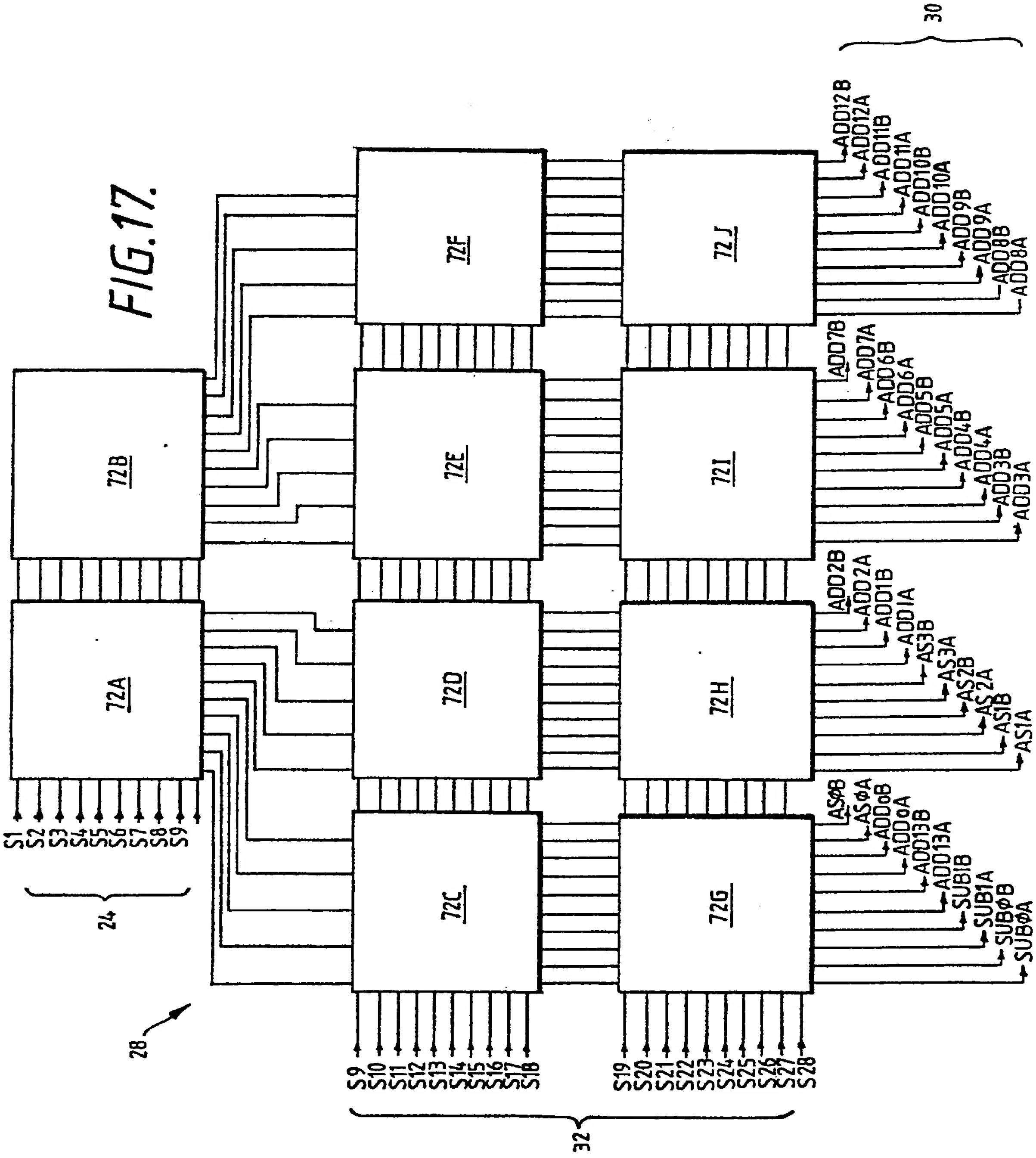

D9
20H
20G
20F
20E
20D
20C
20B
20A
20I
20J
20K
20L
20M
20N
20O
20P
70H
70G
70F
70E
70D
70C
70B
70A
48H
48G
48F
48E
48D
48C
48B
48A
S8
S7
S6
S5
S4
S3
S2
S1
CLR
A
B
Q_H
F164
CLR
A
B
Q_H
F164
CLR
D
Q
1/4
F175
C_o
$\Sigma 2$
A2
B_2
A1
B_1
1/2
F283
$\Sigma 1$
A3
C4
B3
A4
B4
$\Sigma 4$
1/8 F244
1/8 F244
1/8 F244
1/8 F244
22
RESET
X_N
REGCLK
LSB
ADDCLK

FIG.17.


72A
72B
72C
72D
72E
72F
72G
72H
72I
72J
S1
S2
S3
S4
S5
S6
S7
S8
S9
24
28
S9
S10
S11
S12
S13
S14
S15
S16
S17
S18
S19
S20
S21
S22
S23
S24
S25
S26
S27
S28
32
30
ADD12B
ADD12A
ADD11B
ADD11A
ADD10B
ADD10A
ADD9B
ADD9A
ADD8B
ADD8A
ADD7B
ADD7A
ADD6B
ADD6A
ADD5B
ADD5A
ADD4B
ADD4A
ADD3B
ADD3A
ADD2B
ADD2A
ADD1B
ADD1A
AS3B
AS3A
AS2B
AS2A
AS1B
AS1A
ASΦB
ASΦA
ADDΦB
ADDΦA
ADD13B
ADD13A
SUB1B
SUB1A
SUBΦB
SUBΦA

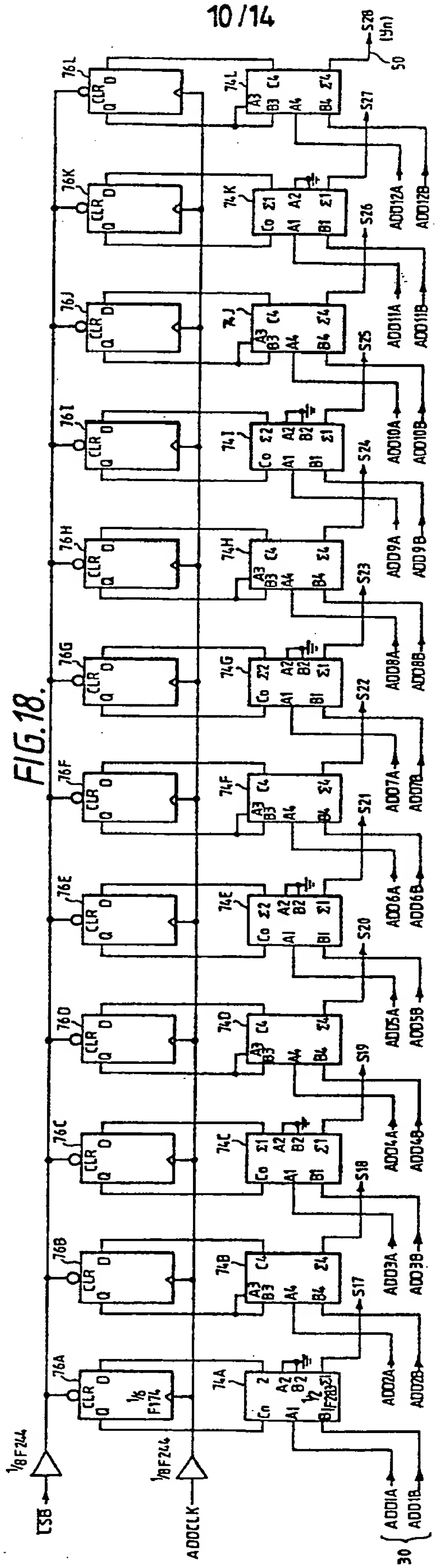
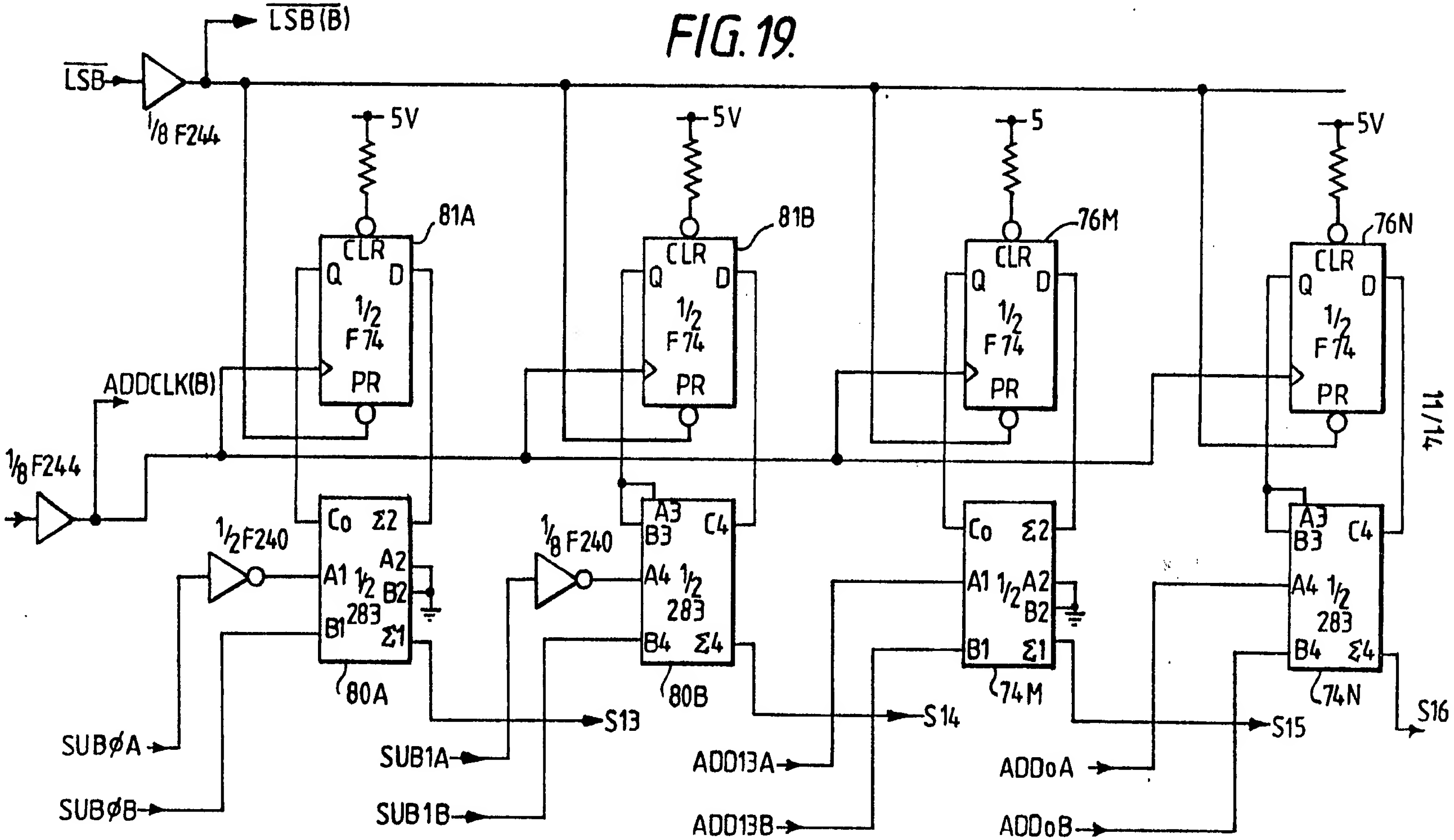

CSB
1/8F244
ADDCLK
1/8F244
76A
CLR
Q
D
1/6
F174
74A
Cn
A1
A2
B2
B1
Σ1
1/2
F283
2
76B
76C
76D
76E
76F
76G
76H
76I
76J
76K
76L
74B
74C
74D
74E
74F
74G
74H
74I
74J
74K
74L
Co
A3
B3
C4
A4
B4
Σ4
Σ2
S17
S18
S19
S20
S21
S22
S23
S24
S25
S26
S27
S28
(Yn)
50
ADD1A
ADD1B
30
ADD2A
ADD2B
ADD3A
ADD3B
ADD4A
ADD4B
ADD5A
ADD5B
ADD6A
ADD6B
ADD7A
ADD7B
ADD8A
ADD8B
ADD9A
ADD9B
ADD10A
ADD10B
ADD11A
ADD11B
ADD12A
ADD12B

FIG.18.

FIG. 19.



LSB
1/8 F244
LSB(B)
5V
5V
5
5V
81A
81B
76M
76N
CLR
Q
D
1/2
F74
PR
ADDCLK(B)
1/8 F244
1/2 F240
1/8 F240
Co
Σ2
A2
B2
A1
B1
1/2
283
Σ1
A3
B3
C4
A4
B4
Σ4
80A
80B
74M
74N
S13
S14
S15
S16
SUBØA
SUBØB
SUB1A
SUB1B
ADD13A
ADD13B
ADDoA
ADDoB

FIG.20.

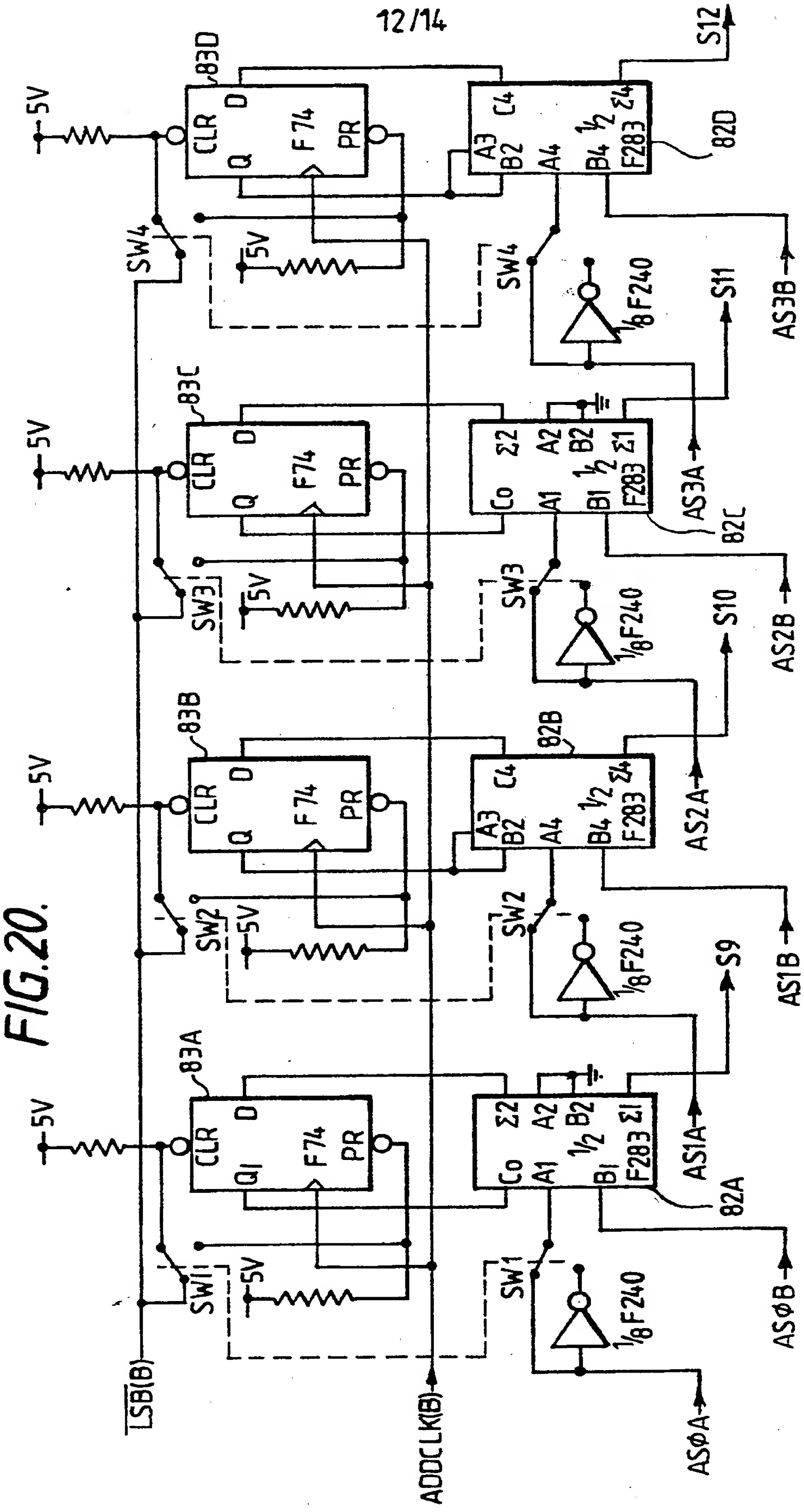
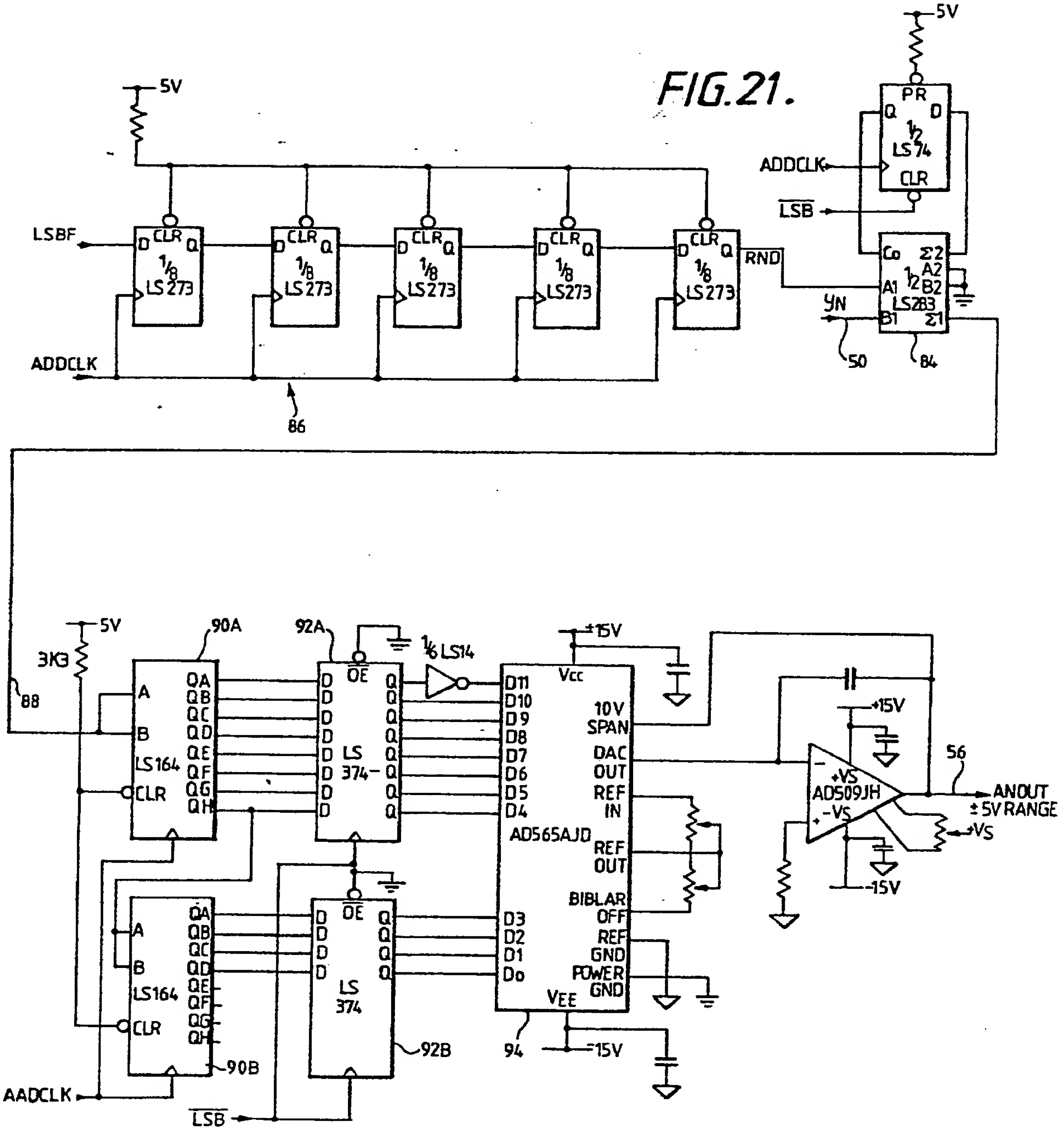

5V
5V
5V
5V
83A
83B
83C
83D
CLR
D
Q1
Q
F74
PR
82A
82B
82C
82D
Co
A1
B1
Σ1
Σ2
A2
B2
½
F283
C4
A3
A4
B4
Σ4
SW1
SW2
SW3
SW4
1/8 F240
$\overline{LSB(B)}$
ADDCLK(B)
ASØA
ASØB
AS1A
AS1B
AS2A
AS2B
AS3A
AS3B
S9
S10
S11
S12
]

FIG.21.


5V
LSBF
D CLR Q
1/8
LS273
ADDCLK
86
RND
5V
PR
Q D
1/2
LS74
ADDCLK
CLR
LSB
Co Σ2
A2
A1 1/2 B2
LS283
YN
B1 Σ1
50
84
88
5V
3K3
90A
A
B
LS164
CLR
QA QB QC QD QE QF QG QH
92A
OE
LS 374
1/6 LS14
±15V
Vcc
D11 D10 D9 D8 D7 D6 D5 D4
10V SPAN
DAC OUT
REF IN
AD565AJD
REF OUT
BIBLAR OFF
D3 D2 D1 Do
REF GND
POWER GND
VEE
94
-15V
+15V
AD509JH
+VS
-VS
-15V
56
ANOUT
±5V RANGE
+VS
90B
92B
AADCLK
LSB

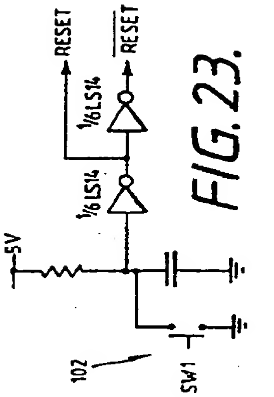

5V
1/6 LS14
1/6 LS14
RESET
$\overline{RESET}$
102
SW1

FIG. 23.

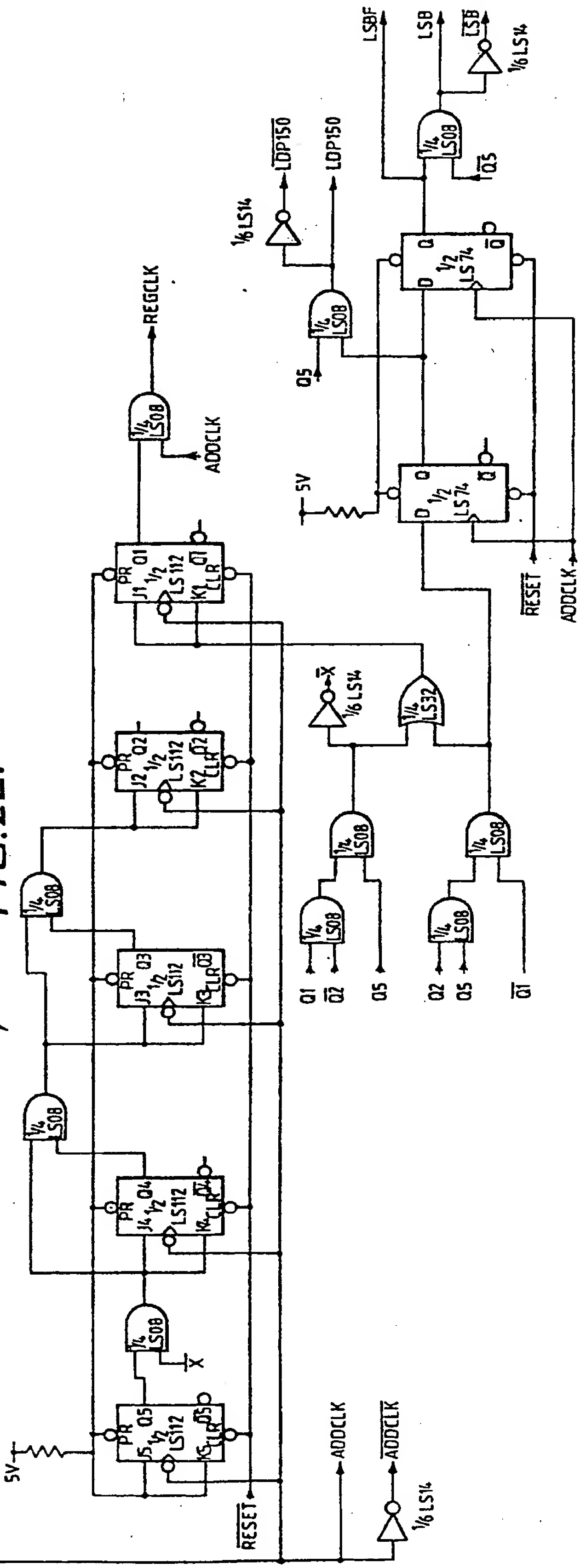

98
5V
VDD
PXO-1000
P1
P2
P3
P4
P5
P6
OUT
RST
VSS
$\overline{RESET}$
1/4 LS08
100
5V
PR
J5
Q5
1/2 LS112
K5
CLR
$\overline{Q5}$
1/4 LS08
X
PR
J4
Q4
1/2 LS112
K4
CLR
$\overline{Q4}$
1/4 LS08
PR
J3
Q3
1/2 LS112
K3
CLR
$\overline{Q3}$
1/4 LS08
PR
J2
Q2
1/2 LS112
K2
CLR
$\overline{Q2}$
PR
J1
Q1
1/2 LS112
K1
CLR
$\overline{Q1}$
1/4 LS08
REGCLK
ADDCLK
$\overline{RESET}$
ADDCLK
$\overline{ADDCLK}$
1/6 LS14
1/4 LS08
1/4 LS08
Q1
$\overline{Q2}$
Q5
1/4 LS08
1/4 LS08
Q2
Q5
$\overline{Q1}$
1/4 LS32
1/6 LS14
$\overline{X}$
5V
D
Q
1/2 LS74
$\overline{Q}$
D
Q
1/2 LS74
$\overline{Q}$
$\overline{RESET}$
ADDCLK
Q5
1/4 LS08
1/6 LS14
$\overline{LDP150}$
LDP150
1/4 LS08
$\overline{Q5}$
LSBF
LSB
1/6 LS14
$\overline{LSB}$

FIG. 22.

2201854

A METHOD AND APPARATUS FOR FILTERING ELECTRICAL SIGNALS

This invention relates to methods of filtering electrical signals and digital filter apparatus.

Digital filtering using finite impulse response (FIR) filters is a technique which has many desirable attributes in the context of high throughput signal processing. These include guaranteed stability, the absence of limit cycles, and linear phase characteristics (if required). In many applications, however, analogue filters are used since digitally implemented filters are too bulky, consume too much power, or, in particular, are too slow to operate in real time.

It is an object of the present invention to widen the application of digital filters, and to this end a method of filtering a time-varying electrical signal according to the invention includes generating a digital signal representative of the time-varying signal, passing the digital signal through a series of delay steps or stages to generate a series of successively delayed signals, and combining the products of the successively delayed signals and a plurality of multiplier coefficients to produce a filtered digital signal by the direct combination of the delayed signals in a succession of addition, subtraction or binary shift operations. By carrying out at least the majority of the required multiplication operations as a series of such primitive operations, the greatest signal processing burden of a digital filter, the multiplication operations, can be considerably reduced. In particular, the series of additions and/or subtractions and/or binary shifts can be arranged so that a number of 'partial sums' or 'partial results' yielded in one coefficient-sample multiplication can be used in other such multiplications. Methods of refining such series are described hereinafter with a view to minimising the number of primitive operators needed to generate a pre-determined set of coefficient-sample products. By reducing the number of primitive operators in this way, the throughput of signals can be increased with reduced filter complexity and, substantially without losing coefficient accuracy. A filter architecture may be formed which comprises in essence an interconnection of elementary operations, i.e. adders, subtractors, and shift elements, together with Z blocks (delay stages).

Non-recursive and recursive digital filters can be constructed in accordance with the invention, in both hardware or software embodiments, or in embodiments which use a combination of hardware and software.

The invention also includes a method of filtering a time-varying electrical signal by generating a digital signal representative of the time-varying signal, generating a plurality of coefficient-sample products by subjecting the digital signal to a plurality of addition and/or subtraction and/or shift operations, the products being represented by a plurality of digital product signals, delaying each product signal with respect to the other product signals, and summing the said delayed product signals to yield a signal representative of a filtered output, the result obtained from at least one coefficient-sample multiplication being used in a further such multiplication or multiplications.

As a further aspect of the invention there is provided a method of filtering one or a plurality of time varying digital signals by generating delayed samples of the said signal or signals, generating a plurality of digital product signals representing a plurality of coefficient-sample products from each of at least one of the samples by subjecting the said samples operations, combining the product signals to generate one or more digital filtered output signals. The partial results generated in the form of at least one of the product signals are used in further coefficient-sample multiplications to economise on the number of primitive operations required. The combination of the said product signals may involve summing groups of said signals to form intermediate output signals which may themselves be combined to form a single output signal using, for example, a further delay element or elements to delay one or more of the intermediate signals relative to each other.

The invention further includes apparatus for carrying out the methods of the invention.

The invention will now be described by way of example with reference to the drawings, in which:-

Figure 1 is a block diagram of a known non-recursive transverse digital filter;

Figure 2 is a block diagram of a transpose of the filter of Figure 1;

Figure 3 is a block diagram of a filter resulting from the combination of the filters of Figures 1 and 2;

Figure 4 is a signal flow graph illustrating the formation of a product with the coefficient 17 from a series of additions;

Figure 5A to 5D are signal flow graphs illustrating successive steps in the formation of several products with a common set of additions so that higher coefficients are derived from lower coefficients in a single process;

Figure 6 is a diagram of a multiplication arrangement for positive and negative coefficients;

Figure 7 is an improved signal flow graph illustrating the formation of the coefficients used in Figures 5A to 5D;

Figures 8A and 8B are diagrams showing the implementation of a binary shift operation in serial and parallel circuitry respectively;

Figure 9 is a signal flow graph similar to the flow graph of Figure 7 but using a greater number of binary shifts and fewer addition or subtraction operations;

Figure 10 is a signal flow graph which is the transpose of the graph of Figure 9;

Figure 11 is a block diagram of a non-recursive digital filter constructed according to the signal flow graph of Figure 10;

Figure 12 is a block diagram of a folded delay line for use in a symmetrical digital filter;

Figure 13 is a simplified block diagram illustrating a filter constructed in accordance with the invention;

Figure 14 is a block diagram showing in more detail a filter of the construction shown in Figure 13;

Figure 15 is a circuit diagram of analogue input circuitry of the filter of Fig. 14;

Figure 16 is a circuit diagram of a folded delay line;

Figure 17 is a circuit diagram of a programmable interconnection area;

Figure 18 is a circuit diagram of an array of adders;

Figure 19 is a circuit diagram of an array of subtractors and adders;

Figure 20 is a circuit diagram of an array of adder/subtractors;

Figure 21 is a circuit diagram of the output stages of the filter of Fig. 14;

Figure 22 is a circuit diagram of the control unit of the filter of Fig. 14; and

Figure 23 is a circuit diagram of a reset circuit.

With reference to Figure 1, a typical conventional transverse type of digital filter structure comprises and input 10, a series of delay stages 12A, 12B, 12N, a plurality of multipliers 14A, 14B, ..., 14N coupled to taps in the line of delay stages to yield the products of delayed signals and respective preselected coefficients a_o, a_1 ..., a_n, and an adder 16 to add all of the products and yield an output signal y_n at output 18. The transpose transverse digital filter structure shown in Fig.2 has corresponding delay stages 13A, 13B, ... 13N, and a plurality of coefficient multipliers 15A, 15B, ... 15N coupled to the input 11 and to a plurality of adders 19A, 19B, ... 19N coupled between the delay stages. A further digital filter structure is shown in Figure 3, resulting from the combination of filter structures of the form shown in Figures 1 and 2. In this case fewer delay line tapping points are required to realise a filter of a given complexity compared with the filters of Figures 1 and 2. The filter is characterised by the generation of a plurality of outputs formed by multiplying a plurality of delayed sample inputs, the outputs then being combined to produce a single output. Thus, the filter has an input 21, a delay line comprising delay elements 23A and 23B, a plurality of coefficient multipliers 25A to 25F which feed a pair of adder blocks 27A and 27B, a further output delay element 29, and a combining adder 31 feeding an output terminal 33. Whether the filters are implemented in hardware or software, or a combination of both, the three main operations are unit delay, addition, and multiplication, with the last of these, multiplication, presenting the greatest computational burden and/or requiring the most logic circuitry. The present invention is directed to reducing the time and/or hardware required for the multiplication step.

The conventional method of performing multiplication is by conditionally adding the multiplicand according to the value of each multiplier bit and then shifting one bit position. For B bit data words, then, conventionally (B-1) shifts are needed and an average (B/2 - 1) additions are needed (the maximum could be B-1). The multiplication operation is seen therefore to be at least a factor B/2 more complex than a simple addition (typically B is in the range 8 to 32). In hardware the multiplication operation will always dominate either in terms of numbers of chips (when made from individual small scale/medium scale integrated circuits) or in terms of chip area (as in dedicated VLSI chips for signal processing). In software the number of instructions required is dominated by the shift and add instructions required for the multiplication.

Whether in hardware or software every multiplication in practice is performed by a collection of elementary operations. Those mentioned already are 'shift' (a left shift corresponds to multiplication by 2) and 'add'. A third elementary operation, 'subtract', can also be added to the list of useful elementary operations. The comparative complexities of these three operations depends on the realisation . In software on a typical processor all three incur the same time penalty. In hardware using parallel data paths the shift operation is almost 'free' since it is a simple path rewire. An adder is of about the same complexity as a subtractor; a combined adder/subtractor is a little more complex than an individual adder or subtractor. In hardware using bit-serial data on a single wire, shift operations are less complex than add/subtract operations but are not now 'free' since they are achieved using a single D-type flip flop.

The type of elementary operation used in practice depends on design decision (particularly in hardware), with one or more of the primitive operator types add/subtract/shift being used.

The methods to be described achieve significant complexity reduction by using fewer elementary operations than do conventional approaches. They produce exactly the same computed results with no approximations.

A non-recursive filter is defined by the convolution operation.

$$y_n = \sum_{i=o}^{N-1} a_i x_{n-i} \qquad (1)$$

where x_n is the input sequence, a_i are the filter coefficients, and y_n is the output sequence. This expression has a realisation in the filter shown in Figure 2, which is the transpose of the filter of Figure 1. In this form it is evident that a single input sample x_n is multiplied independently by each of the N filter coefficients a_i, and summed to form y_n. By letting $x_n.a_o = W_o$, $x_n.a_1 = W_1$, and up to $x_n.a_{N-1} = W_{N-1}$, it will be seen that each of the inner products in (1) above may be formed from a sequence of addition (and/or shift and subtraction) operations. This is most easily demonstrated using signal flow graph notation. As an example, consider the formation of a single product where $a_x = 17$. A possible flow graph is shown in Figure 4, in which each convergence of arrowed lines represents a simple addition. All W_i values may formed independently in this way. However, in doing so, a great deal of redundancy appears in the resulting structure. By allowing any of the intermediate (or partial) sums generated during the formation of any other subsequent product, a single composite flow graph may be constructed which represents the whole of the multiplication part of a filter. This is again best demonstrated by example. E.g. consider a filter with following coefficients:-

$a_o= 1;\ a_1=7;\ a_2=16;\ a_3=21;\ a_4=33$

The filter coefficients may be built up as shown in Figures 5A to 5D thus realising the filter with only eleven addition operations and no shifts. This is to be compared with a minimum of twenty elementary operations if the same coefficients are realised separately in the conventional way. Such savings become increasingly significant with larger filter order and dynamic range.

It should be noted that only positive integer coefficients have been so far considered. If, as is usual, both positive and negative coefficients exist, then each occurrence of a negative value is accommodated by the inclusion of an extra two's complementer (subtractor) at the output (or input) of the multiplication block. This is demonstrated in Figure 6, where W_o, W_1 and W_3 are positive, and W_2 and W_4 are negative. If the filter is specified in terms of fractional coefficients, each coefficient must be appropriately scaled to an integer value.

The example of Figures 5A to 5D illustrates an underlying concept of using partial sums (e.g. 2, 4, 6, 14) formed in one coefficient/sample multiplication process, together, if necessary, with products already formed, to shorten subsequent coefficient/sample multiplication processes.

Thus, by breaking down each inner product in the convolution process into a number of arithmetic operations, one can produce a filter architecture comprising an interconnection of primitive operators and z-blocks (delay stages), offering faster operation and/or reduced complexity.

The construction of a signal flow graph such as that of Figure 5D can be reproduced in a computer program by means of the algorithm ERROR:= COEFF(K) - (COEFF(K-1) + PSUM (i - x)), where the value of ERROR must be non-negative. This process is repeated until

COEFF(K) is exactly formed (i.e. ERROR =0), except that subsequent iterations minimise the value of ERROR: = COEFF(K) - (PSUM(i) + PSUM (i - x)), PSUM(i) being the most recently formed partial sum (the partial sums in Figure 5D being 2, 4, 6, 14, 20, etc,).

The number of operations may be reduced to nine by using an alternative algorithm. First, the coefficients are sorted into ascending order and the error function is initialised to ERROR: = COEFF(K), the next coefficient to be formed. Two existing partial sums are then selected which minimise the function as follows: ERROR: = ERROR - (PSUM(A) + PSUM(B)), where ERROR must always have a non-negative value. This process is repeated until ERROR = 0 (i.e. the coefficient is fully formed). An example of the application of this algorithm is shown in Figure 7, using the same coefficients as for Figure 5D.

In the above algorithm, any occurrence of multiplication by a power of two (i.e. single or repeated addition of the signal to itself may be implemented by replacing the addition operation(s) with a single bit shift(s). The implementation of this operation will vary according to whether the filter is realised using parallel or serial arithmetic. In the case of a serial filter the doubling operation is realised by inserting a single bit delay in the signal path (time shifting), whereas for parallel circuits it simply implies the switching of each signal line to its neighbour of higher binary significance (space shifting). The latter obviously requires no extra hardware and reduces to a routing problem, as shown in Figure 8A (serial) and Figure 8B (parallel).

The advantages apparent for parallel arithmetic may be amplified by employing algorithms which place additional emphasis on the shift operation. Thus, the previous algorithm is modified by allowing each partial sum formed to be repeatedly doubled, thus increasing the number of intermediate results available for future use. This results in a signal flow graph as shown in Figure 9, having four additions and six binary shifts.

The flow graph of Figure 9 can be transposed as shown in Figure 10 (unused binary shifts having been removed), to realise a signal flow graph for constructing a digital filter according to the expression:

$$y_n = x_n + 7x_{n-1} + 16x_{n-2} + 21x_{n-3} + 33x_{n-4}$$

The signal flow graph of Figure 10 yields directly a filter structure as shown in Figure 11, each convergence of a pair of arrowed lines in Figure 10 being represented by an adder in Figure 11.

A symmetrical filter according to expression:-

$$y_n = x_n + 7x_{n-1} + 16x_{n-2} + 21x_{n-3} + 33x_{n-4} + 21x_{n-5} + 16x_{n-6} + 7x_{n-7} + x_{n-8}$$

can be constructed with very few extra adders by folding the delay line and summing outputs having corresponding coefficients, as shown in Figure 12.

If subtraction is included as an available primitive operator, then some improvement in complexity reduction can be achieved.

The error minimisation problem which exists in all the previous algorithms will, in general converge more rapidly (that is with fewer operations in the solution) if both addition and subtraction operations are permitted. The error function with subtraction available is similar to that given above except that it is now only the magnitude of the error that is significant, rather than the magnitude and sign:

ERROR: = ERROR +/- (PSUM(A) +/- PSUM(B))

As with the algorithms referred to above, the coefficients are first sorted into ascending order and starting from unity. It is, however, equally possible to start with all coefficients and converge to unity using a reduction process.

A combination of all three primitive operators addition, subtraction and shift can be used, with the advantages of the shift operation becoming more apparent with parallel digital filter realisations.

The algorithms given above are included purely as examples. It is possible to use several similar algorithms to obtain an efficient filter structure.

Referring to Figure 13, a hardware realisation of a non-recursive filter in accordance with the invention using bit serial arithmetic may be viewed as a tapped delay line 20 having an input 22 and a series of output taps 24, an array of adders/subtractors in an arithmetic block 26, and an interconnection area 28 coupled to the tapped outputs of the delay line and to inputs 30 and outputs 32 of the arithmetic block 26. The interconnection area 28 provides the connection between the delay line 20 and the arithmetic block 26, with the interconnections in the interconnection area being formed according to the filter function required. It should be noted that although the use of primitive operations in the manner described above offers significant savings in numbers of operations the interconnection complexity is increased. For software implementation this would not be important. To overcome this problem for hardware implementation it is possible to map this irregularity onto a regular interconnection area with random connection or switching points. This may be viewed as a programmable function array which may be implemented by a full non-blocking cross-bar switch array, allowing all possible connections between all elements. The area may however be significantly reduced through the use of partial blocking networks.

Referring now to Figure 14, a bit serial non-recursive filter using 'primitive operators' may be constructed generally in accordance with the simplified structure described above with reference to Figure 13.

In addition to the delay line 20, arithmetic block 26 and interconnection area 28, the filter of Figure 14 has an analogue input/output and control unit 34 for converting signals to and from the required bit serial digital format. Thus, the unit 34 has an analogue input 36 which is coupled to an analogue-to-digital (A/D) converter 38 for converting an incoming signal to be filtered to parallel digital samples. Conversion to a series of bit serial samples x_n takes place in a parallel-to-serial converter device 40 having an output coupled to the input 22 of an arithmetic unit 42 comprising the delay block 20 and arithmetic block 26 of Fig 13.

The A/D converter 38 and parallel-to-serial converter 40 are both clocked from control circuitry 44 within the unit 34, this circuitry in turn being clocked by a system reference clock 46.

The delay line 20 is of the folded type already shown in Figure 12, since this particular filter is symmetrical in its transfer function, and has associated with the delay blocks a plurality of 'folding' adders 48 connected in the manner of the adder of Figure 12. The outputs of the adders deliver successively delayed samples x_n to the interconnection area 28 via a plurality of lines 24. It is the function of the interconnection area 28 to take the delayed samples from the delay line 20 and folding adders 48 and to pass samples back and forth to arithmetic devices to build up coefficient-sample products by means of an arrangement similar to the arrangement of adders shown in Figure 11. In this case, the arithmetic devices comprise not only adders 26A, but also subtractors 26B and dual purpose adder/subtractor devices 26C. Each adder, subtractor, and subtractor/adder has an input connected to the interconnection area 28 amongst a bundle 30 of inputs and an output coupled to the interconnection area 28 amongst a bundle 32 of outputs. With each clock interval of the delay line 20, a series of delayed samples reach the arithmetic devices through the interconnection area 28 and the partial sums are built up clock interval by clock interval by successive addition and subtraction operations with the final composite product samples represented by the bit serial digital samples y_n being produced at the output 50 of one of the adders 26A. The signal y_n is now fed to the unit 34 where it is converted to parallel digital format and rounded off in a rounding and serial-to-parallel converter 52 which is, in turn, coupled to an output digital-to-analogue converter 54. The filtered analogue output signal is made available at a filter output 56.

It will be appreciated that the characteristics of the filter are determined by the connections formed in the interconnection area 28 between the delay line 20 and the arithmetic devices 26A, 26B and 26C, and between the arithmetic devices themselves. As mentioned above, these interconnection devices may be programmable cross-bar switch devices to allow the filter characteristics to be altered. Indeed, the filter could be made dynamically adaptive by constructing the interconnection area as a dynamically variable array of switches. In many applications, however, the filter characteristic will be fixed and the cost and bulk of programmable connection devices could be avoided by simple hard-wired connections.

Examples of circuitry for the blocks of Figure 14 are shown in Figures 15 to 23.

Referring to Figure 15 the input stages of the filter comprise a buffer amplifier 60, a level-setting arrangement 62, and an 8-bit A/D converter chip 38. Parallel-to-serial conversion is carried out by shift register 64 and an output flip-flop 66 to yield a digital bit-serial signal x_n at output 68. From this output, the digitised signal is fed to the delay line input 22, as seen in Figure 16. The delay line 20 is folded, as in Figure 14, and has 16 TTL logic delay blocks constituted by shift register integrated circuits 20A to 20P type No 74F164 or 74LS164, output pairs such as the outputs of circuits 20B and 20N, or 20C and 20M, being coupled to adder i.c.'s 48A to 48H clocked froma clock input ADDCLK via a D-type flip-flops, 70A to 70H. The adder outputs 51 to 58 form the output taps 24 of Figure 14 coupled to eight inputs of the programmable inter-connection array 28, shown in more detail in Figure 17.

Referring to Figure 17, the interconnection area in this embodiment is formed by ten programmable 10 x 10 cross-bar switch matrices 72A to 72J available from Radiospares Ltd, under type No.RS 468-024. Each matrix allows programmable connections from each X-line to be made to one or more than one Y-line. Thus the output taps of the delay line 20 may each be connected to any of the Y-lines connecting matrices 72A and 72B, through the other eight matrices, to a proportion of the outputs, 30 of the interconnection area 28. Referring back to Figure 14, it will be seen that the outputs 30 feed the input of the series of adders 26A, subtractors 26B, and combined adder/subtractors 26C. These arithmetic units are shown in detail in Figure 18, 19 and 20 respectively. Since the addition operation is most commonly required, there are fourteen adders 74A to 74N each clocked by D-type flip-flops 76A to 76N (and formed by 4-bit parallel adders type 74LS283), whereas there are only two similarly constructed subtractors 80A and 80B and four adder/subtractors 82A to 82D, clocked by D-type flip-flops 81A, 81B and 83A to 83D. All three groups of arithmetic units feed back to the X-inputs 32 of matrices 72C to 72J of the interconnection area 28 (see Figure 17), except for the final adder 74L, the output of which carries the basic digital output of the filter y_n on line 50 (Figures 14 and 18).

Referring to Figure 21, conversion of the digital output y_n back to analogue form is preceded by a rounding stage comprising an adder i.c. 84 driven by a chain of flip-flops 86 clocked by the control circuitry. The rounded bit-serial signal is then fed on line 88 to a serial-to-parallel converter comprising two 8-bit shift registers 90A and 90B type nos. 74LS164,and multiple flip-flop chips 92A and 92B, to feed a 12-bit parallel signal to a D/A converter i.c. 94 type no AD565AJD. Finally, the analogue output signal generated at the output DACOUT is conducted to the output 56 of the filter via a differential buffer amplifier 96.

All timing for the digital parts of the filter is carried out by the control circuitry of Fig. 22. This includes a switchable clock oscillator 98, and a JK flip-flop and AND-gate chain 100, which together provide the clock REGCLK for the delay line shift registers, ADDCLK for the arithmetic units, A/D converter and serial/parallel and parallel/serial converters, and the least significant bit time slot marker (LSB) input for the arithmetic units. The reset circuit shown in Figure 23 allows the control unit to be reset by means of a manual reset button 102.

CLAIMS:

1. A method of filtering an electrical input signal or signals, comprising combining a plurality of samples x_a, x_b, x_c, ... of a digital signal or signals representing the input signal or signals to yield an output signal y_n which is a function of ax_a, bx_b, cx_c, ... , where a, b, c, ... are multiplying coefficients, wherein the combination comprises a succession of addition, subtraction and/or binary shift operations at least some of which each serve to form signals associated with more than one of the products ax_a, bx_b, cx_c, ...

2. A method according to claim 1, including the generation of signals forming partial or intermediate combinations of at least some of the samples, which combinations serve as the inputs for addition, subtraction and/or binary shift operations each associated with the generation of a signal associated with more than one of the said products.

3. A method according to claim 1 or claim 2, wherein the input signal is a time-varying signal, wherein x_a, x_b, x_c, ... are all the same signal, and wherein the method includes the combination of product signals representing the products ax, bx, cx, ... to form y_a y_b, y_c, ... respectively which combination includes delaying the product signals with respect to each other.

4. A method according to claim 1 or claim 2, wherein x_a, x_b, x_c, ... are samples of the digital signal or signals which are delayed with respect to each other, and wherein y_n is the sum of the products ax_a, bx_b, cx_c, ...

5. A method according to claim 1 or claim 2, wherein the majority of operations in the generation of the output signal y_n from the samples x_a, x_b, x_c, .. are primitive addition, subtraction, binary shift and/or delay operations.

6. A method of filtering a time-varying electrical signal comprising subjecting a digital signal representative of the time-varying signal to a series of delay steps or stages to generate a series of successively delayed signals, generating an output signal representative of a combination of products of the delayed signals and a plurality of multiplier coefficents by the direct combination of the delayed signals in a succession of addition, subtraction and/or binary shift operations.

7. A method of filtering a time-varying electrical signal comprising generating a plurality of coefficent-sample products by subjecting a digital signal representative of the time-varying signal to a plurality of addition, subtraction, and/or binary shift operations, the products being represented by a plurality of digital product signals, delaying the product signals with respect to each other, and summing the mutually delayed product signals to yield a signal representing a filtered output.

8. A method of filtering one or a plurality of time-varying signals by generating digital signals representing the time-varying signal or signals, generating delayed samples of the said digital signals, generating a plurality of product signals representing a plurality of coefficient sample products from the or each of at least one of the samples by subjecting the said samples to a plurality of addition, subtraction and/or shift operations, combining the product signals

to generate one or more digital filtered output signals.

9. A method according to any of claims 6,7 or 8, wherein a partial result or results generated in at least one coefficient-sample multiplication is or are used in another coefficient-sample multiplication or multiplications.

10. A method according to any of claims 6 to 9, wherein the combination of the product signals involves summing groups of the said signals to form intermediate output signals which are themselves combined to form a single output signal using a further delay element or elements to delay one or more of the intermediate signals relative to each other.

11. A method according to any preceding claim, wherein the arrangement of the addition, subtraction, and/or binary shift operations embodies or is derived from a directed acyclic graph in which internal nodes are freely interconnected.

12. A method according to any of claims 1 to 10, wherein the arrangement of the addition, subtraction, and/or binary shift operations is based on a repeated optimisation procedure involving obtaining coefficients by selecting the said operations which produce, from a starting coefficient and any intermediate coefficients already produced, coefficients closest to a wanted multiplying coefficient.

13. Apparatus for filtering an electrical input signal including a multiplication arrangement having one or a plurality of digital signal inputs for receiving the said input signal or signals and one or a plurality of

outputs for making available one or a plurality of output signals which are a function of products of the input signal or signals and a plurality of multiplier coefficients, wherein the arrangement includes a plurality of adders, subtractors and/or binary shift elements.

14. Apparatus according to claim 13, whrein each of at least some of the adders, subtractors and/or binary shift elements generates signals which are associated with more than one of the said products.

15. Apparatus according to claim 13 or claim 14, further comprising a chain of delay elements having taps coupled either to a plurality of inputs or to a plurality of outputs of the multiplication arrangement.

16. Data processing apparatus programmed to perform a filtering method according to any of claims 1 to 12.

17. A method of filtering an electrical signal, the method being substantially as herein described with reference to the drawings.

18. Apparatus for filtering an electrical signal, the apparatus being constructed and arranged substantially as herein described with reference to the drawings.

Published 1988 at The Patent Office, State House, 66/71 High Holborn, London WC1R 4TP. Further copies may be obtained from The Patent Office.